Neural Computing and Applications
https://doi.org/10.1007/s00521-019-04468-3

ORIGINAL ARTICLE

Deep learning for spoken language identification: Can we visualize speech signal patterns?

Himadri Mukherjee[1] · Subhankar Ghosh[5] · Shibaprasad Sen[6] · Obaidullah Sk Md[2] · K. C. Santosh[3] · Santanu Phadikar[4] · Kaushik Roy[1]

Received: 26 May 2019 / Accepted: 26 August 2019

Abstract
Western countries entertain speech recognition-based applications. It does not happen in a similar magnitude in East Asia. Language complexity could potentially be one of the primary reasons behind this lag. Besides, multilingual countries like India need to be considered so that language identification (words and phrases) can be possible through speech signals. Unlike the previous works, in this paper, we propose to use speech signal patterns for spoken language identification, where image-based features are used. The concept is primarily inspired from the fact that speech signal can be read/visualized. In our experiment, we use spectrograms (for image data) and deep learning for spoken language classification. Using the IIIT-H Indic speech database for Indic languages, we achieve the highest accuracy of 99.96%, which outperforms the state-of-the-art reported results. Furthermore, for a relative decrease of 4018.60% in the signal-to-noise ratio, a decrease of only 0.50% in accuracy tells us the fact that our concept is fairly robust.

Keywords Language identification · Spectrogram · Speech pattern · Convolutional neural network

1 Introduction

The field of speech recognition [1–4] has evolved largely ever since Dudley's attempt of computerized speech recognition in 1930 [5]. Speech recognizers in different Western languages like English are now commercially available [6]. The development of speech recognizers has not been that much for the Indic languages, and thus, the residents of the South Asian countries have not been able to take complete advantage of the advents of speech recognition. It is due to the fact that the complexity of the Indic languages is itself very challenging to model which is aggravated by the multilingual nature of these countries. People here seldom use a mixture of languages while talking, and thus, it is very important to detect the language of the spoken words and phrases prior to recognition.

✉ K. C. Santosh
santosh.kc@usd.edu

Himadri Mukherjee
himadrim027@gmail.com

Subhankar Ghosh
sgcs2005@gmail.com

Shibaprasad Sen
shibubiet@gmail.com

Obaidullah Sk Md
sk.obaidullah@aliah.ac.in

Santanu Phadikar
sphadikar@yahoo.com

Kaushik Roy
kaushik@wbsu.ac.in

[1] Department of Computer Science, West Bengal State University, Kolkata, India

[2] Department of Computer Science and Engineering, Aliah University, Kolkata, India

[3] Department of Computer Science, The University of South Dakota, Vermillion, SD, USA

[4] Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, Kolkata, India

[5] CVPR Unit, Indian Statistical Institute, Kolkata, India

[6] Department of Computer Science and Engineering, Future Institute of Engineering and Management, Kolkata, India

Published online: 05 September 2019

Automatic language identification is the technique of automatically identifying the language from speech signals. Such a system can aid toward speech recognition in multilingual countries like India by determining the language of the spoken segments to invoke the language-specific recognizers.

Tang et al. [7] attempted language identification on four languages, namely Bangla, Gregorian, Turkish and Assamese, from the Babel corpus. They used phonetic information coupled with recurrent neural network (RNN)-based approach and obtained better results than the baseline system. Giwa et al. [8] studied the effect of language identifiers on the performance of speech recognizers. Their study involved the use of two datasets, namely the SADE corpus and the Multipron corpus. Experiments were performed on four languages, namely Afrikaans, Sesotho, English and isiZulu. They experimented with three scenarios namely single language tag, multilingual tag and all four language tag. A recall value of 100% was obtained for the all four tag scenarios on both the datasets. Gunawan et al. [9] identified five languages in the thick of English, Malay, Korean, Chinese and Arabic. They used mel frequency cepstral coefficient (MFCC) features coupled with vector quantization on a database put together with the help of 10 volunteers. They obtained individual accuracies of 90%, 80%, 80%, 60% and 100% for the aforementioned sequence of languages, respectively.

Masumura et al. [10] experimented with parallel phonetically aware DNNs and long short-term memory recurrent neural networks to enhance language identification performance. They experimented with twelve languages from the GlobalPhone database. They used a 38-dimensional feature comprising of 12 MFCC, their deltas and double deltas. They reported utterance error rate of 0.48 and frame error rate of 14.56 on the test set. He et al. [11] presented a system to differentiate Uyghur and Kazakh with audio of short texts containing less than fourteen words each. They used heuristic-based features coupled with a maximum entropy-based classifier and obtained precisions of 96.5% and 94.6% for Kazakh and Uyghur, respectively. Jin et al. [12] experimented with a senone-based approach for language identification from the NIST LRE 2009 dataset for twenty-three languages. Their system outperformed state-of-the-art methods involving deep neural networks and i-vectors when the senones were used for classification as well as formation of i-vector. Mukherjee et al. [13] differentiated Tamil, Telugu, Malayalam and Kannada with line spectral pair-grade (LSP-G) features and fuzzy classification. The features involved extraction of linear predictive coefficients which were used to generate the LSPs. The band-wise grades were computed from these which served as features. They worked with over 12,000 clips, and 96.46% accuracy was obtained. They had also experimented with phoneme recognition [6] for phoneme-based language identification. At the outset, they attempted to distinguish the seven Bangla vowel phonemes and reported an accuracy of 98.35% with MFCC and neural networks. Gupta et al. [14] experimented with six languages, namely Malayalam, Marathi, Telugu, Bangla, Tamil and Hindi from the IIIT-H dataset. They used LPC and MFCC features along with random forest and SVM classifier. A highest accuracy of 92.6% was obtained using random forest and a combination of both the features. Madhu et al. [15] presented a system for distinguishing seven Indic languages in the thick of Hindi, Bangla, Urdu, Telugu, Manipuri, Punjabi and Assamese. Their experiment involved the use of both prosodic information and phonotactic features. An artificial neural network was trained with 2 h of data per language. They reported accuracies of 68% and 72% for the prosodic and phonotactic information-based systems, respectively.

Masumura et al. [10] experimented with parallel phonetically aware DNNs and long short-term memory recurrent neural networks to enhance language identification performance. They experimented with twelve languages from the GlobalPhone database. They used a 38-dimensional feature comprising of 12 MFCC, their deltas and double deltas. They reported utterance error rate of 0.48 and frame error rate of 14.56 on the test set. Nercessian et al. [16] experimented with telephonic and narrowband broadcast speech in nine languages. They used deep neural network (DNN) bottleneck features with i-vectors and combined their other proposed methodologies and obtained 30% improvement over the baseline DNNs.

Rebai et al. [17] attempted out of set data identification involved in open set language identification. They experimented with the LDC2015E88 and LDC2015E87 datasets and obtained 6% relative decrease in equal error rate by using deep support vector machine (SVM) over classical methodologies. Berkling et al. [18] used phoneme-based information for identifying three languages in the thick of English, German and Japanese. They reported an accuracy of 84.1% with just 15 features for distinguishing Japanese and English.

Srivastava et al. [19] used RNNs for modeling language-based phonotactic information. Their convex combination of statistical and recurrent neural network outperformed standard DNN-based systems in terms of mean $F1$ score for more than 176 languages. Tang et al. [20] used phonetic features from DNN-based system along with RNN for language identification. They experimented with the Babel and AP16-OLR datasets and reported better results compared to standard acoustic neural models. Their system also outperformed standard i-vector-based systems for short and noisy audio. Mukherjee et al. [21] applied lazy learning coupled with a newly

proposed features named MFCC-2 for language identification. The feature involved approximation of MFCCs. This was followed by band grades to model the energy distribution. The final metrics were mean and standard deviations for each of the bands. They worked with unconstrained data as well as numeral data in three languages, namely English, Bangla and Hindi. Their dataset consisted of over 40,000 clips, and a highest accuracy of 98.09% was reported. They had also experimented with feature reduction techniques like PCA as well. In another instance, they distinguished the three languages from numerals [22]. Their data consisted of recordings using various devices and sources. They used MFCC-based features coupled with neural network-based classifier and reported an accuracy of 98.39%. Watanabe et al. used convolutional neural networks (CNNs) coupled with bidirectional long short-term memory (LSTM) as well as RNN-based language model for language identification. They performed multifarious experiments which are detailed in [23]. They reported average character error rates of 16.6 and 21.4 for seven and ten languages, respectively. Revathi et al. [24] used perceptual feature for distinguishing seven Indian languages in the thick of Kannada, Bangla, Hindi, Marathi, Malayalam, Telugu and Hindi. They experimented with different setups and reported a highest accuracy of 99.4% with a clustering-based technique.

Zissman et al. [25] proposed four different approaches for language identification including Gaussian mixture model and phoneme-based approaches. They experiment with the OGI multilingual telephonic speech corpus and obtained accuracies of 94.5% and 79.2% for two and ten language closed sets, respectively. Zissman [26] also applied phoneme recognition coupled with phonotactic language modeling on the same dataset and obtained an accuracy of 89% for 11 language closed set. Highest two language classification accuracies of 98% for 45-s long segments and 95% for 10-s long segments were also obtained in this experiment. Saikia et al. [27] studied the effect of language-independent transcribers for language identification. They experimented with eight Indian languages, namely, Telugu, Gujarati, Tamil, Mizo, Manipuri, Hindi, Bangla and Assamese. Their dataset comprised of both studio recorded data and real-world YouTube data. They concluded the fact that transcription-based classification performed better than audio-feature-based classification. Lamel et al. [28] experimented with phone recognition in English and French. They experimented with the BREF, Wall Street journal and TIMIT corpus. They found it easy to distinguish French at phone level than at the lexical level and reported phone error rate of 23.6 on BREF. They also reported accuracy of over 99% for language identification with 2-s long clips.

Ghozi et al. [29] have used a visual approach with the aid of MFCC features for audio scene analysis. Their approach involved the use of inter-similarity amidst the frames. Dennis [30] has talked about disparate image processing techniques which can be applied to sound event recognition. He has also outlined the different type of features that can be extracted from spectrograms. Montalvo et al. [31] used spectrogram textures for segregating five languages in the thick of English, Russian, French, Mandarin and Spanish. They reported a lowest equal error rate of 4.8% on fusing i-vector-based representation with their proposed technique. The various available works in the literature are summarized in Table 1.

2 Key contributions

The key contributions of our work are presented as follows:

- The system distinguishes seven Indic languages from the standard dataset (The IIIT-H Indic Speech Databases [32]). The system segregates the languages by modeling their frequency envelope textures. This is very much challenging as because different languages have common word syllables which present a commonality in the frequency envelopes. Our system visualizes such textures for classifying the languages.
- We have made use of deep learning with the aid of CNNs to characterize the seven Indic languages. Our experimental outcome shows the effectiveness of the proposed method as compared to reported works in the literature as well as other popular textural and audio-based features.
- The robustness of the proposed system in different noisy conditions was tested as well and for a relative decrease of 4018.60% in the signal-to-noise ratio (SNR), a decrease of only 0.50% in the system accuracy was obtained. This makes our system suitable for outdoor conditions.
- Experiments were performed with standard deep learning architecture and other datasets as well for comparative study, and we obtained better results. The datasets were larger in terms of both size and the number of languages and also had noisy data.

In the rest of the paper, the proposed methodology is presented in Sect. 3 followed by the details of experimental setup in Sect. 4. The results are presented in Sect. 5 and the conclusion in Sect. 6. The proposed methodology is diagrammatically illustrated in Fig. 1.

Table 1 Summary of available works in the literature

References	Languages	Technique highlight	Results
Tang et al. [7]	Assamese, Bangla, Turkish and Georgian	Phonetic information and RNN	2.92% (error rate, max)
Giwa et al. [8]	Afrikaans, English, Sesotho and isiZulu	JSM-based approach with voting	100/% (recall)
Gunawan et al. [9]	Arabic, Chinese, English, Korean and Malay	MFCC and vector quantization	78 % (accuracy)
Gupta et al. [14]	Hindi, Bangla, Tamil, Telugu, Malayalam and Marathi	LPC, MFCC with random forest and SVM	92.6% (accuracy)
He et al. [11]	Uyghur and Kazakh	Heuristics and entropy-based classification	95.1% (accuracy)
Jin et al. [12]	Amharic, Cantonese, Bosnian, Creole, Dari, Croatian, American-English, Indian-English, Farsi, French, Georgian, Hausa, Hindi, Korean, Mandarin, Pashto, Portuguese, Russian, Spanish, Turkish, Ukrainian, Urdu and Vietnamese	Senone-based approach	1.41% (error rate)
Madhu et al. [15]	Bangla, Hindi, Telugu, Urdu, Assamese, Punjabi and Manipuri	Prosodic and phonotactic information	72% (accuracy)
Masumura et al. [10]	French, Korean, German, Mandarin, Portuguese, Russian, Shanghai, Spanish, Swedish, Thai, Turkish, Vietnamese	MFCC	0.48% (error rate)
Nercessian et al. [16]	Cantonese, Hindi, Farsi, Mandarin, Korean, Russian, Urdu, Spanish and Vietnamese	DNN with i-vectors	2.42% (error rate)
Saikia et al. [27]	Assamese, Manipuri, Bangla, Hindi, Tamil, Gujarati, Telugu and Mizo	Transcription-based classification	96% (max accuracy)
Tang et al. [20]	Assamese, Bangla, Georgian, Cantonese, Pashto, Turkish and Tagalog	Phonetic features	2.40% (error rate)
Watanabe et al. [23]	English, Japanese, Mandarin, German, Spanish, French, Italian, Dutch, Russian, Portuguese	CNN + LSTM	100% (max accuracy)
Zissman et al. [25]	English, Farsi, French, German, Japanese, Korean, Mandarin, Spanish, Tamil and Vietnamese	GMM and others	79.2% (accuracy)
Zissman [26]	English, Farsi, French, German, Japanese, Korean, Mandarin, Spanish, Chinese, Tamil and Vietnamese	Phonotactic language modeling	79% (accuracy)
Berkling et al. [18]	German, Japanese and English	Phoneme-based information	84.1% (accuracy)
Lamel et al. [28]	French and English	Phone recognition	Over 99% (accuracy)
Montalvo et al. [31]	English, Spanish, Mandarin, Russian and Spanish	Spectrogram + i-vector	4.8% (error rate)

3 Proposed method

3.1 Spectrogram

Real-life sound signals are composed of a set of disparate frequency components each of which are present in different proportions in a sound signal. It is this proportion of the components along with their frequencies which determines what we actually hear. This set is often called as spectral envelope. The shape of this envelope plays a dominant part in what we hear. Spectrograms are a way to represent this envelope. It is composed of data having three dimensions. The x-axis denotes the time interval of a sound clip, while the y-axis denotes the frequencies (the range of frequencies present). The energy of a particular frequency component at a particular time is denoted by means of color codes which is the third dimension. Spectrograms can be in grayscale as well as in red–green–blue (RGB) format.

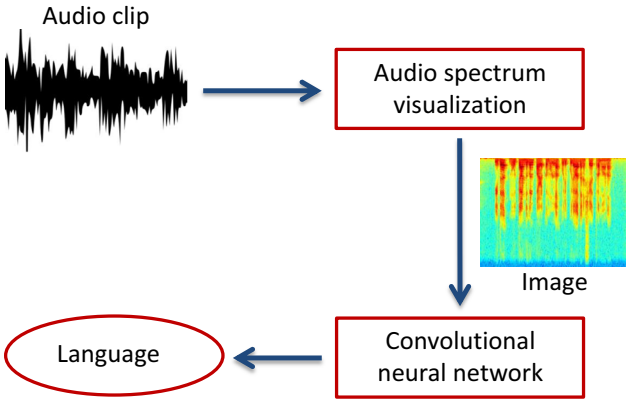

Fig. 1 The proposed methodology

The color intensity corresponds to the energy of a frequency component at a particular instance.

In order to compute the spectrogram from an audio signal, it is divided into small sections or frames each of which are subjected to Fourier transformation. This technique is often referred to as short-term Fourier transform. The ultimate result is obtained by joining the individual Fourier transformations. A trade-off is also involved in this technique amidst time and frequency resolution. A clip when divided into shorter segments aids in better time resolution but weaker frequency resolution. When the size of clips is increased, the frequency resolution improves but the time resolution degrades. In the present experiment, we have worked with both longer and shorter sub-clips. We had split the clips into frames of sizes 128, 256, 512 and 1024 sample points which are denoted as F_1, F_2, F_3 and F_4, respectively. Frames of having a size which is a perfect power of 2 facilitate in Fourier transformation without the need of padding. The obtained spectrograms for the 256 point frame (best result) are shown in Fig. 2.

The dataset was also subjected to three types of noises by adding wind sound, fan sound and aircraft cabin sound to the clips. The details of the signal-to-noise ratio values are presented in Sect. 5.1.

3.2 Deep learning-based classification with convolutional neural network

Deep learning [33] is a technique which involves the use of multilayer learning models for learning patterns within data with disparate levels of abstraction at each layer. The input data are processed at each layer and are passed on to the next. Deep learning-based neural networks differ from standard neural networks primarily in the number of layers as well as the type of processing between the input and output layers. There are different types of deep networks such as recurrent neural networks, long short-term memory networks, deep belief networks and CNNs [34]. Deep learning is being widely used in handling multimedia data as well [35].

CNNs [36–38] are one of the most popular type of deep neural networks which have a profound advantage in

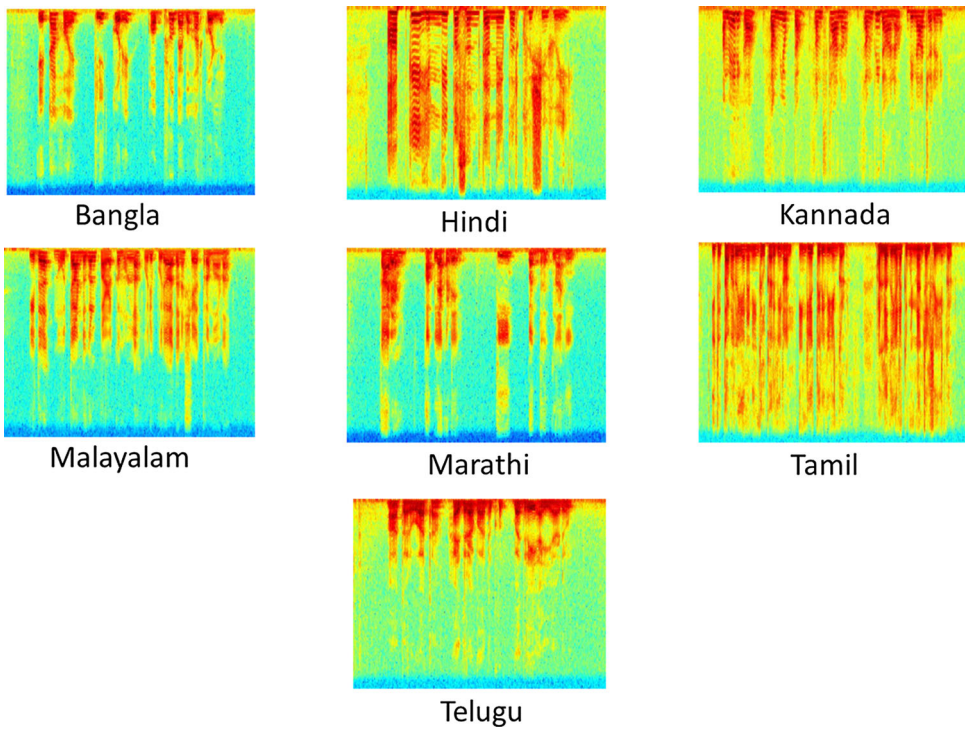

Fig. 2 Spectrograms for the seven different languages

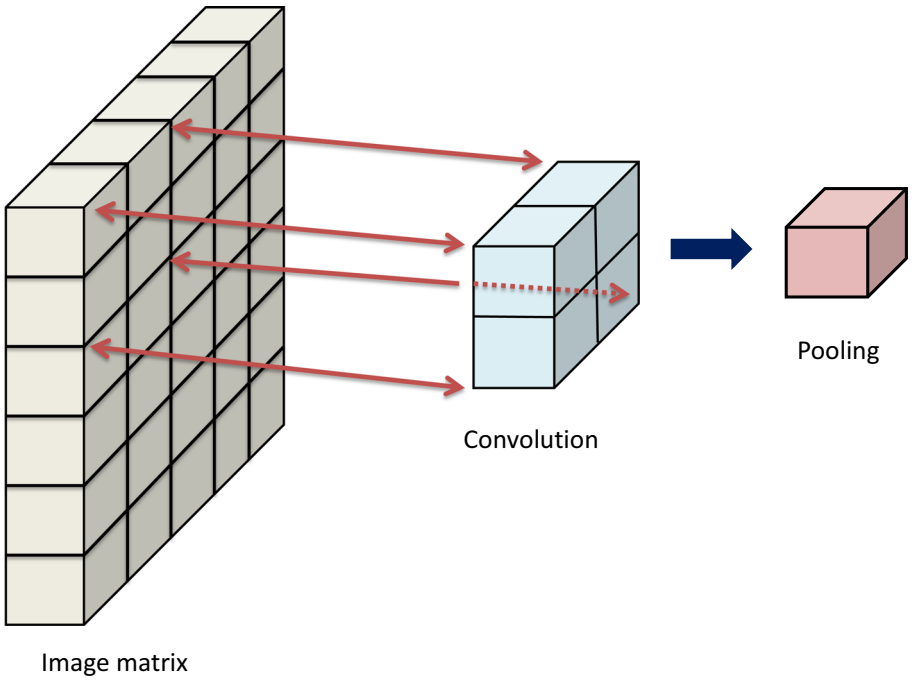

Fig. 3 Structure of a minimal convolutional neural network

handling spatial distribution of data [34]. It consists of convolution and pooling layers in the initial phase. The final layer is a fully connected layer also known as dense layer whose dimension corresponds to the number of output classes. The structure of a minimal convolutional neural network is graphically illustrated in Fig. 3. During convolution, the original matrix is multiplied with the filter in an element-wise manner, and then, the sum of the product is considered. The convolution technique is presented in Algorithm 1.

Algorithm 1: Convolution

Input : IMG

Output: IMG_conv

```
for i ← 0 to n − csize do
    for j ← 0 to n − csize do
        for k ← i to i + csize do
            for l ← j to j + csize do
                IMG_conv(k, l) ← (IMG(k, l) ∗ CONV((k − i), (l − j));
                l ← l + 1;
            end
            k ← k + 1;
        end
        j ← j + stride;
    end
    i ← i + stride;
end
```

Prior to calculation of the next element, the second important facet is introduced, known as stride. The stride marks the number of pixels which overlap between two successive convolutions. A stride of one, after the first convolution would involve the elements at ($O(0, 1)$, $O(0, 2)$, $O(1, 1)$, $O(1, 2)$) in generation of the second element of the convoluted matrix. The shift of the filter across the original matrix with respect to stride value is presented in Fig. 4.

Post-convolution, pooling is performed. This can be thought of as a dimension-reducing procedure, wherein an element from the convolution matrix for each pooling window is chosen. The max pooling procedure is graphically illustrated in Fig. 5 wherein a stride value of one is used.

The layers in the end are mostly fully connected layers which are known as dense layers. The dimension of the dense layers may be varied, but the final dense layer must have a dimension which is same as the number of output classes.

We used a CPCPDD architecture where C implies convolution layer, P implies pooling layer and D implies dense layer. The dimension of the first and second convolution layers was set to five and three, while the max pooling size was varied from two to five used. The size of the first dense layer was set to 256, while the final dense layer had to have a dimension of 7 corresponding to the number of output classes. Four sizes of 50×50, 100×100, 150×150 and 200×200 for scaling the images were used in the experiment. The intermediate layers used a ReLU activation (as shown in Eq. 1), while the final dense layer used a softmax activation (as shown in Eq. 2). The architecture of the CNN along with the different parameters was chosen after trial runs.

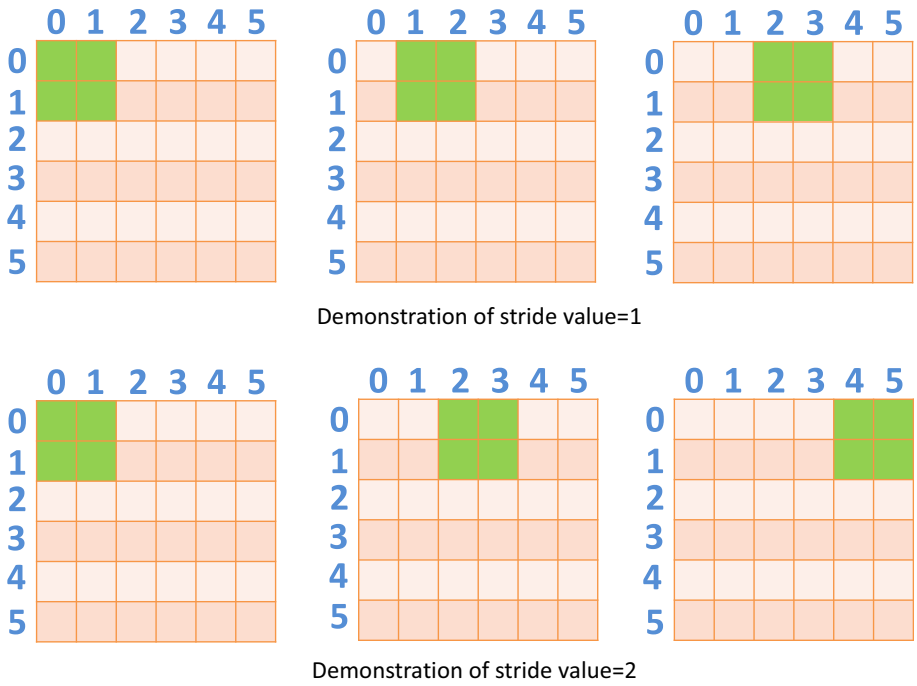

Fig. 4 Shift of convolution window across an image for different stride values

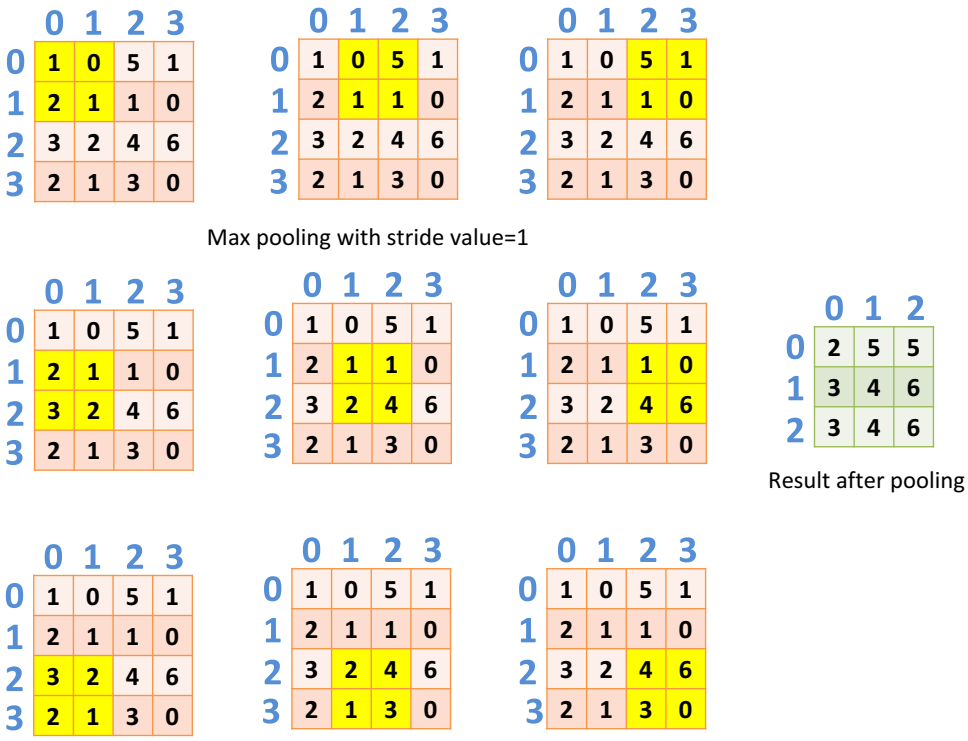

Fig. 5 Max pooling for different stride values

$$f(x) = \max(0, x), \tag{1}$$

here x is the input to a neuron.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \tag{2}$$

where z is an input vector of length K.

The proposed CNN architecture is diagrammatically illustrated in Fig. 6.

3.3 Training with GPU

The architecture of the present-day graphical processing units (GPUs) allows parallel processing up to a large

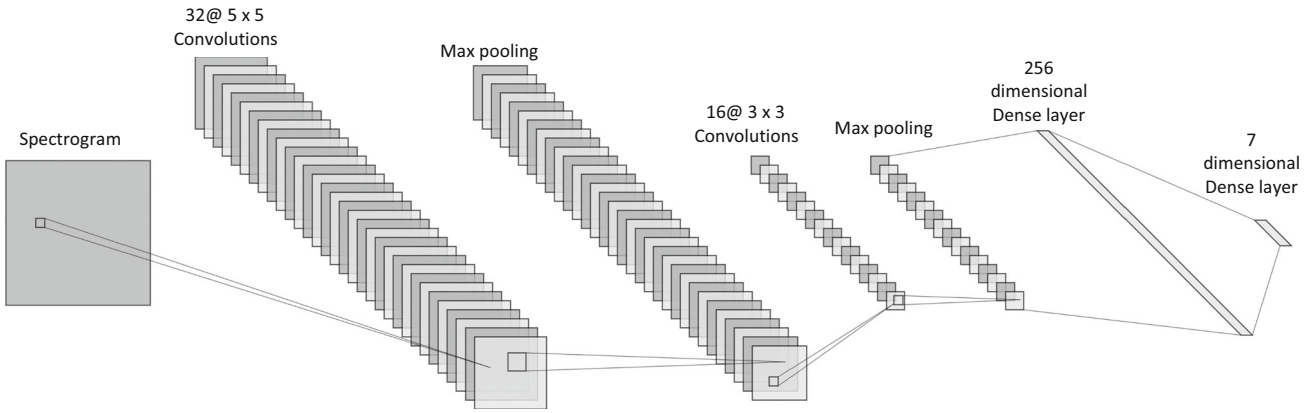

Fig. 6 Architecture of CNN used in our proposed technique

Table 2 Number of trainable parameters of the different layers of our proposed CNN architecture

Layer	Parameters
Convolution 1	2432
Pooling 1	0
Convolution 2	4624
Pooling 2	0
Dense 1	409,856
Dense 2	1799
Total	418,711

extent. This is very helpful in processing large volumes of data. Our proposed architecture comprised of a high number of trainable parameters as presented in Table 2. The pooling layers do not have any parameters to train as because they contribute only to selection. We made use of a Nvidia Quadro P3000 GPU with 6 GB memory to handle the huge number of parameters of the CNN during the course of our experiments.

4 Experimental setup

4.1 Dataset

Data are a very important aspect of an experiment. The database should uphold real-world characteristics so that the developed system is robust. In this paper, we have experimented with the database of seven Indic languages [32], namely Marathi, Bangla, Hindi, Tamil, Telugu, Malayalam and Kannada. The database has been developed by speech and vision laboratory, IIIT Hyderabad. These languages were considered in the database because native speakers of these languages were readily available and the number of articles in these languages was over 10,000.

The database consists of 1000 sentences for each of the languages. Each of the sentences is available as a separate clip in the database. These sentences spanned over 5000 most frequently used words in the text of the corresponding languages. The audios were recorded in a studio with a microphone, which was connected to a zoom handy recorder. Along with the audio clips, the text data of the recordings are also present in IT3 transliteration scheme as well as UTF-8 format. The duration of data for each of the languages is presented in Table 3.

4.2 Evaluation metric and protocol

In our present experiment, we have used a k-fold cross-validation technique [22] for evaluating the system. This scheme ensures that every instance with a dataset is subjected to training as well as testing at least once. In this method, a dataset is subdivided into k parts out of which $(k-1)$ parts are used for training and a single part is used for testing. This is repeated k times. The value of k was set to 5 as presented in [39]. The classification accuracy can be computed as:

$$\text{Accuracy} = \frac{\text{Correctly classified instances}}{\text{Total instances}} * 100. \quad (3)$$

5 Results and analysis

5.1 Results using the proposed technique

The obtained accuracies for F_1–F_4 with the initial setup of the CNN are presented in Table 4. The confusion matrix for F_2 is presented in Table 5.

Table 3 Details of the dataset

Language	Marathi	Hindi	Telugu	Malayalam	Kannada	Tamil	Bangla
Duration (HH:MM)	1:56	1:12	1:31	1:37	1:41	1:28	1:39

Table 4 Accuracies for F_1–F_4 with the initial setup of the CNN

Set name	F_1	F_2	F_3	F_4
Accuracy (%)	99.79	99.96	99.70	99.60

It is observed from Table 5 that out of seven classes, five classes were identified with 100% accuracy. Since F_2 produced the best result, we further experimented with it to obtain performance improvement.

It is also seen that two of the clips of Malayalam were classified as Tamil. It was found that the both the Malayalam and Tamil clips had several common keywords like "Malayalam," "Hindi," "India," etc. Among the two misclassified clips, one had both "Hindi" and "Malayalam," while the other had "Malayalam" and "India." The presence of these keywords aided to the confusion. The class-wise distribution of different evaluation metrics is presented in Table 6.

It is also seen that the only other misclassified instance was a single Marathi clip, which was classified as Bangla. It was observed that the Marathi clip had the keyword "Engreji" meaning "English" which was present in different Bangla clips as well which aided to the confusion.

The size of the resized images was experimented with whose result is presented in Table 7. It is seen that the best result was obtained for 100-dimensional images. When the dimension of the resized image was both increased, the accuracy dropped due to overfitting. The accuracy dropped as well when the image size was diminished, thereby pointing to underfitting. This is mainly due to the loss of important textural properties on further diminishing the image.

The pooling size was also varied from 2 to 5 whose results are presented in Table 8. It is observed that the best result was obtained when the size of pooling was set to 3. Further on increasing the pool size, the accuracy dropped which points toward the fact that higher value of pooling led to loss of important data.

The output of the different layers for a single instance from our dataset is presented in Fig. 7. In every stage, the color image is segregated into its corresponding red (R), green (G) and blue (B) plates. Each of these plates is processed individually and then overlapped.

Table 5 Confusion matrix for F_2

	Bangla	Marathi	Telugu	Tamil	Malayalam	Kannada	Hindi
Bangla	1000	0	0	0	0	0	0
Marathi	1	999	0	0	0	0	0
Telugu	0	0	1000	0	0	0	0
Tamil	0	0	0	1000	0	0	0
Malayalam	0	0	0	2	998	0	0
Kannada	0	0	0	0	0	1000	0
Hindi	0	0	0	0	0	0	1000

Table 6 Class-wise evaluation metrics using CNN

	Sensitivity	Specificity	Precision	Negative predictive value	False-positive rate	False discovery rate	False-negative rate	Accuracy	*F*1 score
Bangla	0.999	1	1	1	0	0	0.001	1	1
Marathi	1	1	0.999	1	0	0.001	0	1	0.999
Telugu	1	1	1	1	0	0	0	1	1
Tamil	0.998	1	1	1	0	0	0.002	1	0.999
Malayalam	1	1	0.998	1	0	0.002	0	1	0.999
Kannada	1	1	1	1	0	0	0	1	1
Hindi	1	1	1	1	0	0	0	1	1

Table 7 Results for different resize dimensions

Dimension of resized image	50 × 50	100 × 100	150 × 150	200 × 200
Accuracy (%)	98.91	99.96	99.69	99.49

Table 8 Results for different pooling sizes

Pooling size	2	3	4	5
Accuracy (%)	99.83	99.96	99.04	93.39

5.2 Performance in noisy condition

The dataset was also subjected to three types of noises in the thick of wind sound, fan sound and aircraft cabin sound. The average language-wise signal-to-noise ratios for each of the noise sources are presented in Table 9.

The obtained spectrograms for these noise sources are presented for some of the languages in the thick of Hindi, Bangla and Kannada in Fig. 8. The three noise sources can be ordered in the following manner based on the signal-to-noise ratios: wind, fan and aircraft. It can be observed from Fig. 8 that the original information in the clips was distorted significantly in the noisy scenarios. The network structure along with the other parameters was selected based on the best-performing setup on the original dataset. The confusion matrix for the wind noise scenario is presented in Table 10. It can be seen that Marathi and Hindi were recognized with 100% accuracy, while the least accurate result was obtained for Bangla. This is mostly due to the similarity of frequency components of different sections in some Bangla clips with that of the noise components. An overall accuracy of 99.51% was obtained for this scenario, which is the highest among all the three noise scenarios.

Table 9 Signal-to-noise ratios for the different languages

Language	Aircraft	Wind	Fan
Bangla	− 11.31	6.77	− 7.84
Marathi	− 18.04	− 0.30	− 14.53
Hindi	− 21.36	− 3.14	− 17.90
Telugu	− 19.03	− 0.74	− 15.58
Malayalam	− 20.72	− 2.66	− 17.27
Kannada	− 14.85	3.15	− 11.38
Tamil	− 18.69	− 0.07	− 15.23
Average	− 17.71	0.43	− 14.25

The confusion matrix for the fan noise scenario is presented in Table 11. Here also, it can be observed that Hindi was recognized with 100%. It can also be observed that though the SNR decreased by nearly 3213.95%, the average accuracy reduced to 98.6% showing a decrease of only 0.91%. This proves the ability of our system to handle noisy scenarios effectively.

The SNR value was further reduced to − 17.71 in the case of aircraft noise, which is the most noisy scenario in our experiment, whose confusion matrix is presented in Table 12. It can be observed that both Hindi and Telugu were recognized with 100% accuracy. As compared to the least noisy dataset, a decrease of only 0.50% was observed for this dataset; however, the decrease in signal-to-noise ratio was almost 4018.60%, which is extremely

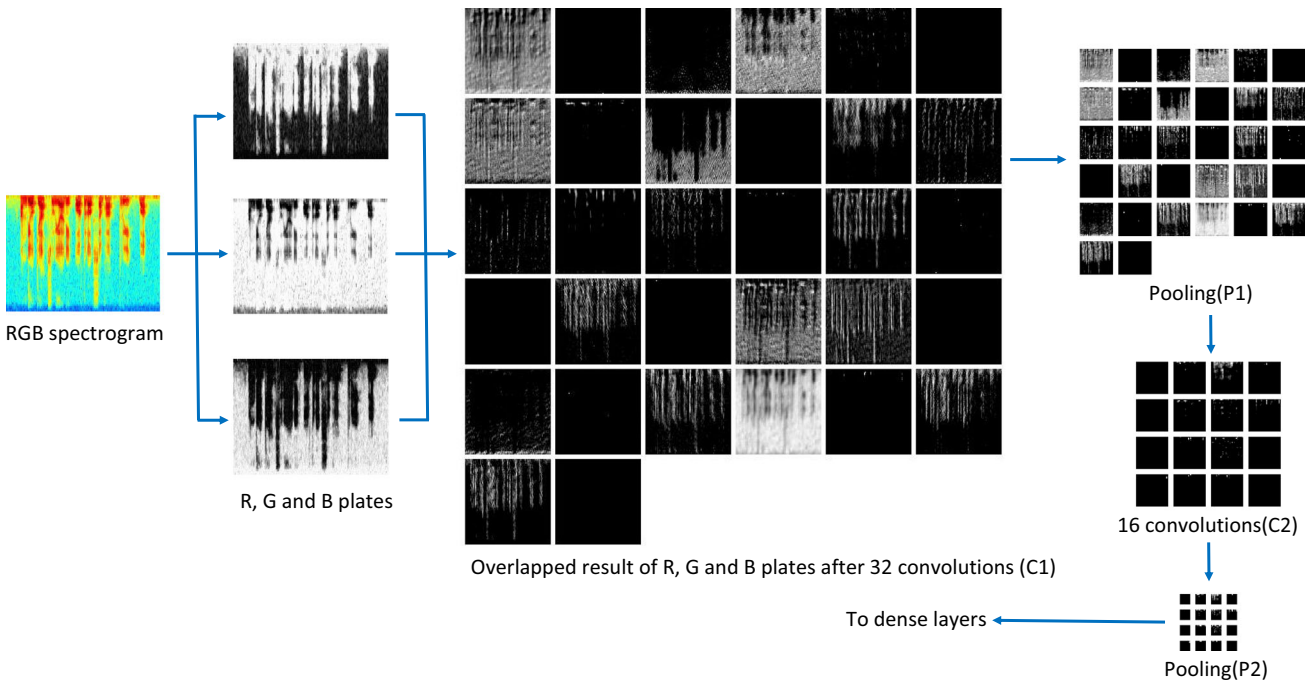

Fig. 7 Output of the different layers of our proposed CNN architecture for a Malayalam clip

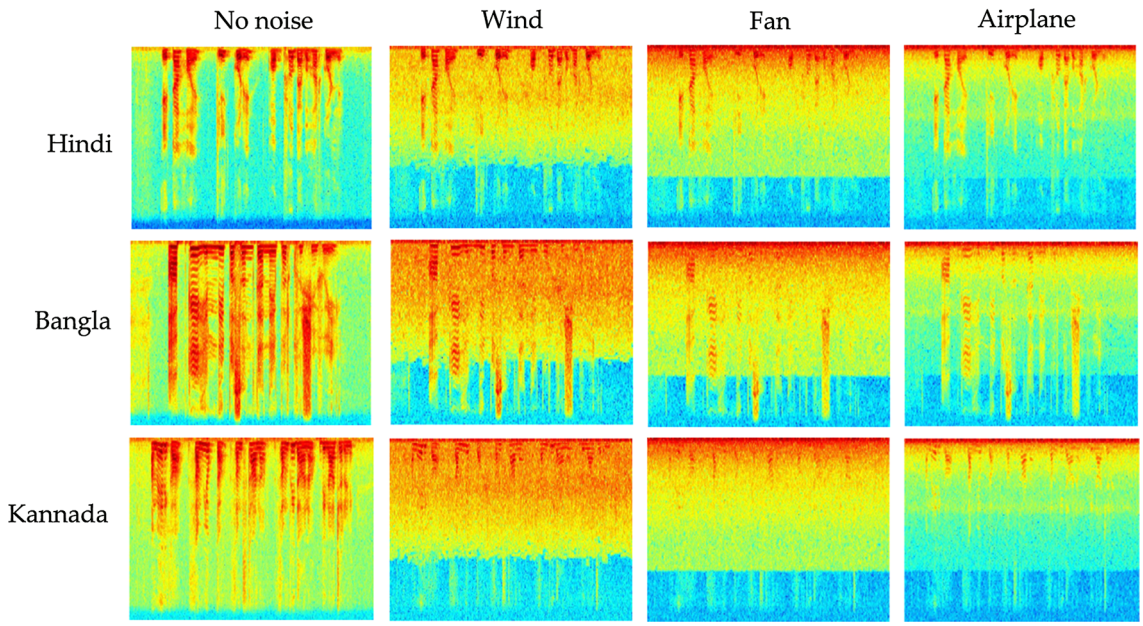

Fig. 8 Spectrograms of Hindi, Bangla and Kannada in no-noise condition as well as in wind noise, fan noise and aircraft cabin noise

Table 10 Confusion matrix for wind noise

	Bangla	Marathi	Telugu	Tamil	Malayalam	Kannada	Hindi
Bangla	986	4	1	5	0	4	0
Marathi	0	1000	0	0	0	0	0
Telugu	0	0	996	2	0	2	0
Tamil	2	1	0	993	0	0	4
Malayalam	0	2	0	2	995	1	0
Kannada	2	1	1	0	0	996	0
Hindi	0	0	0	0	0	0	1000

Table 11 Confusion matrix for fan noise

	Bangla	Marathi	Telugu	Tamil	Malayalam	Kannada	Hindi
Bangla	968	9	2	15	0	6	0
Marathi	0	972	2	22	0	4	0
Telugu	0	0	992	0	8	0	0
Tamil	0	9	0	989	1	0	1
Malayalam	0	0	0	8	992	0	0
Kannada	2	1	5	1	1	989	1
Hindi	0	0	0	0	0	0	1000

Table 12 Confusion matrix for airplane noise

	Bangla	Marathi	Telugu	Tamil	Malayalam	Kannada	Hindi
Bangla	976	7	1	8	1	6	1
Marathi	0	993	4	2	0	1	0
Telugu	0	0	1000	0	0	0	0
Tamil	2	11	1	985	1	0	0
Malayalam	0	1	0	3	996	0	0
Kannada	0	1	18	0	0	981	0
Hindi	0	0	0	0	0	0	1000

Table 13 Class-wise measure of performance metrics in the presence of airplane noise

	Sensitivity	Specificity	Precision	Negative predictive value	False-positive rate	False discovery rate	False-negative rate	Accuracy	*F*1 score
Bangla	0.998	0.996	0.976	1	0.004	0.024	0.002	0.996	0.987
Marathi	0.98	0.999	0.993	0.997	0.001	0.007	0.02	0.996	0.987
Telugu	0.977	1	1	0.996	0	0	0.023	0.997	0.988
Tamil	0.987	0.997	0.985	0.998	0.003	0.015	0.013	0.996	0.986
Malayalam	0.998	0.999	0.996	1	0.001	0.004	0.002	0.999	0.997
Kannada	0.993	0.997	0.981	0.999	0.003	0.019	0.007	0.996	0.987
Hindi	0.999	1	1	1	0	0	0.001	1	1

Table 14 Performance of different classifiers after reduction

	C/I/S	PCA
BayesNet	99.59	95.91
Naïve Bayes	96.96	93.94
LibLINEAR	99.53	99.49
LibSVM	99.63	99.71
MLP	99.89	99.80
RBF network	99.01	97.44
Random Forest	99.84	99.24

encouraging. Various popular performance metrics were computed for this setup, which is presented in Table 13.

5.3 Performance of reduction techniques

We had also experimented with different reduction techniques from Weka [40] on the LSF-based features (best result among audio and other image-based features). The reduction techniques included Pearson's correlation-based reduction (C), principal component analysis (PCA), information gain-based reduction (I) and symmetric uncertainty-based reduction (S). It was observed that the accuracy for the different classifiers was same across all the reduction techniques except PCA, which is presented in Table 14.

5.4 Comparison

5.4.1 Results using standard textural features

Three well-known textural features are used for a comparison.

1. *Gray-level co-occurrence matrix (GLCM)* [41] It is a statistical technique of examining texture with the aid of spatial relationship of pixels. In this technique, the texture of an image is modeled by calculating the frequency of occurrence of different pixel pairs with a certain value and in a specified spatial relationship.
2. *Local binary patterns (LBP)* [42] In this technique, an image is subdivided into cells having a fixed number of pixels. Next, each of the pixels in a cell is compared to all its 8 neighboring pixels. Finally, a normalized histogram of all the cells is computed which is the feature vector for the image.
3. *Weber local descriptor (WLD)* [43] This is a simple technique which works on the principle that humans perceive patterns not only with the change but also with the intensity of a stimulus. It is composed of two aspects, namely orientation (Θ) and differential excitation (ξ). The differential excitation is a function of the ratio between the relative intensity differences between a given pixel x_c with its neighboring pixels x_i. The orientation corresponds to the intensity of the given pixel. The differential excitation and orientation are defined as:

$$\xi(x_c) = arctan\left[\sum_{i=0}^{n-1}\left(\frac{x_i - x_c}{x_c}\right)\right]. \quad (4)$$

For these features, we used classifiers, such as multilayer perceptron (MLP), SVM, BayesNet, Naïve Bayes, RBF, random forest and LibLINEAR. The obtained results are presented in Table 15.

Table 15 Performance of different classifiers on texture-based features

	GLCM	LBP	WLD
BayesNet	49.54	72.51	54.94
Naïïve Bayes	47.37	73.06	55.11
LibLINEAR	62.37	81.5	62.9
SVM	58.56	68.8	48.63
MLP	88.1	94.87	88.43
RBF Net	50.69	78.01	59.41
Random Forest	73.79	89.5	81.46

Table 16 Confusion matrix best-performing textural feature (LBP)

	Bangla	Marathi	Telugu	Tamil	Malayalam	Kannada	Hindi
Bangla	903	31	7	16	3	39	1
Marathi	17	926	17	32	4	4	0
Telugu	3	20	972	5	0	0	0
Tamil	11	51	9	915	12	0	2
Malayalam	10	5	0	7	975	2	1
Kannada	34	3	2	2	2	955	2
Hindi	2	0	0	1	1	1	955

Table 17 Different performance metrics per language for the best-performing textural feature (LBP)

	Sensitivity	Specificity	Precision	Negative predictive value	False-positive rate	False discovery rate	False-negative rate	Accuracy	*F*1 score
Bangla	0.921	0.983	0.903	0.987	0.017	0.097	0.079	0.974	0.912
Marathi	0.894	0.987	0.926	0.981	0.013	0.074	0.106	0.973	0.91
Telugu	0.965	0.995	0.972	0.994	0.005	0.028	0.035	0.991	0.969
Tamil	0.936	0.985	0.915	0.989	0.015	0.085	0.064	0.978	0.925
Malayalam	0.978	0.996	0.975	0.996	0.004	0.025	0.022	0.993	0.976
Kannada	0.954	0.992	0.955	0.992	0.008	0.045	0.046	0.986	0.955
Hindi	0.994	0.999	0.995	0.999	0.001	0.005	0.006	0.998	0.994

Table 18 Confusion matrix with MFCC-2 feature-based system

	Bangla	Marathi	Telugu	Tamil	Malayalam	Kannada	Hindi
Bangla	998	1	1	0	0	0	0
Marathi	0	1000	0	0	0	0	0
Telugu	1	0	999	0	0	0	0
Tamil	1	0	42	957	0	0	0
Malayalam	0	0	1	0	992	0	7
Kannada	0	0	0	0	0	1000	0
Hindi	2	0	12	0	0	0	986

Table 19 Confusion matrix using system proposed in [13]

	Bangla	Marathi	Telugu	Tamil	Malayalam	Kannada	Hindi
Bangla	641	321	2	2	11	1	22
Marathi	245	729	3	0	9	1	13
Telugu	0	28	948	17	5	0	2
Tamil	0	0	41	921	16	3	19
Malayalam	5	24	14	12	889	30	26
Kannada	0	0	0	0	6	994	0
Hindi	40	32	15	37	38	5	833

It is observed from Table 15 that MLP produced the best result for all the three feature sets, while the lowest accuracy for GLCM was obtained with Naïve Bayes. The lowest accuracies for both LBP and WLD were obtained with SVM. Among the three features, LBP produced the best result, whose confusion matrix is presented in Table 16. Different performance metrics were computed for this setup and are listed in Table 17.

5.4.2 Results using standard audio-based features

The performance of two standard identification systems was available in the literature which audio-based features were also applied on the dataset. The first system [21] works with MFCC-2 features and lazy learning. The same framework was used in our experiment on the seven language datasets which produced an accuracy of 99.03%.

Table 20 Class-wise performance metrics for MFCC-2-based system [21]

	Sensitivity	Specificity	Precision	Negative predictive value	False-positive rate	False discovery rate	False-negative rate	Accuracy	*F*1 score
Bangla	0.996	1	0.998	0.999	0	0.002	0.004	0.999	0.997
Marathi	0.999	1	1	1	0	0	0.001	1	1
Telugu	0.947	1	0.999	0.991	0	0.001	0.053	0.992	0.972
Tamil	1	0.993	0.957	1	0.007	0.043	0	0.994	0.978
Malayalam	1	0.999	0.992	1	0.001	0.008	0	0.999	0.996
Kannada	1	1	1	1	0	0	0	1	1
Hindi	0.993	0.998	0.986	0.999	0.002	0.014	0.007	0.997	0.989

Table 21 Language-wise comparison of the proposed system with the reported works

	Marathi	Telugu	Tamil	Malayalam	Kannada	Hindi	Bangla	Total
Revathi et al. [24]	99.22	99.44	98.89	99.67	98.67	100	99.78	99.38
Gupta et al. [14]	95.01	91.67	91.54	92.05	–	90.02	93.86	92.60
Our method	99.90	100	100	99.80	100	100	100	99.96

Table 22 Confusion matrix using Mobilenet

	Bangla	Marathi	Telugu	Tamil	Malayalam	Kannada	Hindi
Bangla	1000	0	0	0	0	0	0
Marathi	0	1000	0	0	0	0	0
Telugu	0	0	1000	0	0	0	0
Tamil	2	1	2	995	0	0	0
Malayalam	0	1	0	2	997	0	0
Kannada	2	1	0	3	0	994	0
Hindi	2	0	2	0	0	1	999

Table 23 Class-wise values of performance metrics for Mobilenet

	Sensitivity	Specificity	Precision	Negative predictive value	False-positive rate	False discovery rate	False-negative rate	Accuracy	*F*1 score
Bangla	0.996	1	1	0.999	0	0	0.004	0.999	0.998
Marathi	0.997	1	1	0.999	0	0	0.003	1	0.999
Telugu	0.998	1	1	1	0	0	0.002	1	0.999
Tamil	0.995	0.999	0.995	0.999	0.001	0.005	0.005	0.999	0.995
Malayalam	1	0.999	0.997	1	0.001	0.003	0	1	0.998
Kannada	0.999	0.999	0.994	1	0.001	0.006	0.001	0.999	0.996
Hindi	1	1	0.999	1	0	0.001	0	1	0.999

The confusion matrix for this setup is presented in Table 18.

The second system [13] works with fuzzy classification and LSP-G features. We had replicated this setting as well, which produced an accuracy of 85.07%. The inter-class confusions are shown in Table 19.

Different performance metrics are presented in Table 20 for the MFCC-based system [21] which produced the best result among the audio-based features.

5.4.3 Comparison with the reported works

The language-wise performance of our system as compared to the one presented by Revathi et al. [24] is presented in Table 21. We observe that our system outperforms the system of Revathi et al. [24]. We have also compared our system with the one proposed by Gupta et al. [14]. However, they experimented with only six languages of the dataset as shown in Table 21.

Table 24 Obtained accuracies (A) and ranks (R) for disparate parts of the dataset

Classifiers	Parts of the dataset					Mean rank
	#1	#2	#3	#4	#5	
CNN						
A	100	99.93	99.86	99.93	100	2.4
R	(3)	(2)	(2.5)	(2.5)	(2)	
Naïve Bayes						
A	98.86	98.93	97.5	97.79	98.14	7.5
R	(7.5)	(6)	(8)	(8)	(8)	
RBF Net						
A	98.86	98.64	98.43	99.07	99.14	7.1
R	(7.5)	(8)	(6)	(7)	(7)	
SVM						
A	100	99.93	99.86	99.93	100	2.4
R	(3)	(2)	(2.5)	(2.5)	(2)	
MLP						
A	100	98.86	99.86	100	100	3.1
R	(3)	(7)	(2.5)	(1)	(2)	
Lib						
A	100	99.64	98.93	99.79	99.86	4.5
R	(3)	(5)	(5)	(5)	(4.5)	
RF						
A	100	99.93	99.86	99.86	99.86	3.2
R	(3)	(2)	(2.5)	(4)	(4.5)	
BayesNet						
A	98.93	99.79	97.71	99.71	99.49	5.8
R	(6)	(4)	(7)	(6)	(6)	

5.5 Performance of established deep learning architecture

To compare the performance of our system, the dataset was subjected to Mobilenet [44], which is an established and standard CNN architecture. An accuracy of 99.79% was obtained with this architecture whose confusion matrix is presented in Table 22. It is observed from Table 22 that the recognition accuracies for Kannada and Tamil decreased as compared to our proposed architecture. Different popular performance metrics were computed for this setup and are presented in Table 23.

5.6 Statistical significance test

The nonparametric Friedman test [45] was performed in order to check for statistical significance. The experiments were done with the noisy dataset which was partitioned into five parts (N). Each of these parts was subjected to eight classifiers (k). The classifiers were then ranked based on their accuracies. The obtained accuracies and their corresponding ranks are presented in Table 24.

The Friedman statistic (χ_F^2) [45] was calculated with the aid of Table 24 in accordance with the following:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{5}$$

where R_j represents the mean rank of a classifier. The critical value of (χ_F^2) at a significance (α) of 0.05 for the specified values of N and k was found to be 14.067. We obtained a value of 24.933 which differed significantly from the critical value, thereby rejecting the null hypothesis:

Further, Iman et al.'s statistic [45] (F_F) was also computed in accordance with the under mentioned:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}. \tag{6}$$

The critical value of this test for α value of 0.05 was found to be 2.359. We obtained a value of 9.907 for (F_F), which also differed significantly from the critical value, thereby rejecting the null hypothesis as well.

Nemenyi's test [45] was carried out as per post hoc test for pairwise comparison of the classifiers. Two classifiers can be considered to be significantly different based on performance if their average ranks differ by at least the critical difference (CD) calculated using

$$\mathrm{CD} = q_\alpha\sqrt{\frac{k(k+1)}{6N}}. \tag{7}$$

The value of $q_{0.05}$ and $q_{0.10}$ for eight classifiers in the case of Nemenyi's test was found to be 3.031 and 2.780 [45] which produced CDs of 4.70 and 4.31, respectively. The calculated CD values for all the classifier pairs are presented in Table 25.

Bonferroni–Dunn [45] test was further performed to compare the performance of CNN (control classifier) along with the others. The computational and evaluation procedure of Bonferroni–Dunn's test is similar to that of Nemenyi's test. The two differ in the values for $q_{0.05}$ and $q_{0.10}$ which are 2.690 and 2.450 leading to CD values of 4.17 and 3.80, respectively. The calculated CD values for the classifier pairs are presented in Table 26.

5.7 Results on other datasets

To check the performance of our system both in terms of robustness and in terms of accuracy, experiments were performed with two datasets. The first dataset was an extension of the dataset used in our experiments. It had over 70 h of data from the top-11 languages used in India [46]. The languages embodied were English, Bangla,

Table 25 Results of Nemenyi's test for all the classifier pairs

CNN								
Naïve Bayes	5.1							
RBF	4.7	0.4						
LibSVM	0	5.1	4.7					
MLP	0.7	4.4	4	0.7				
LibLINEAR	2.1	3	2.6	2.1	1.4			
RF	0.8	4.3	3.9	0.8	0.1	1.3		
BayesNet	3.4	1.7	1.3	3.4	2.7	1.3	2.6	
	CNN	Naïve Bayes	RBF	LibSVM	MLP	LibLINEAR	RF	BayesNet

Table 26 CD values for Bonferroni–Dunn's test with CNN as the control classifier

Classifiers	Naive Bayes	RBF	LibSVM	MLP	LibLINEAR	RF	BayesNet
CD	5.1	4.7	0	0.7	2.1	0.8	3.4

Table 27 Obtained accuracies using our proposed technique for the top-11 languages spoken in India

	English	Bangla	Hindi	Telugu	Tamil	Marathi	Malayalam	Kannada	Urdu	Gujarati	Odia
English	2187	5	27	14	1	4	14	0	78	22	32
Bangla	7	2303	48	4	6	1	13	1	10	62	3
Hindi	20	19	2269	3	4	1	52	0	20	8	10
Telugu	24	37	34	2298	1	0	16	0	1	0	11
Tamil	7	2	11	0	2373	0	5	0	34	13	17
Marathi	0	0	0	0	0	2440	0	0	17	0	0
Malayalam	19	21	36	59	15	1	2221	20	22	12	2
Kannada	0	0	0	0	0	0	2	2447	0	0	0
Urdu	26	7	24	0	2	45	12	0	2296	6	6
Gujarati	11	12	7	5	9	0	20	6	7	2247	5
Odia	20	11	3	11	0	0	2	1	7	0	2374

Table 28 Class-wise performance metrics for the top-11 languages spoken in India

	Sensitivity	Specificity	Precision	Negative predictive value	False-positive rate	False discovery rate	False-negative rate	Accuracy	*F*1 score
English	0.942	0.992	0.917	0.994	0.008	0.083	0.058	0.987	0.93
Bangla	0.953	0.993	0.937	0.995	0.007	0.063	0.047	0.99	0.945
Hindi	0.923	0.994	0.943	0.992	0.006	0.057	0.077	0.987	0.933
Telugu	0.96	0.995	0.949	0.996	0.005	0.051	0.04	0.991	0.954
Tamil	0.984	0.996	0.964	0.998	0.004	0.036	0.016	0.995	0.974
Marathi	0.979	0.999	0.993	0.998	0.001	0.007	0.021	0.997	0.986
Malayalam	0.942	0.991	0.915	0.994	0.009	0.085	0.058	0.987	0.928
Kannada	0.989	1	0.999	0.999	0	0.001	0.011	0.999	0.994
Urdu	0.921	0.995	0.947	0.992	0.005	0.053	0.079	0.987	0.934
Gujarati	0.948	0.996	0.965	0.995	0.004	0.035	0.052	0.992	0.956
Odia	0.965	0.998	0.977	0.996	0.002	0.023	0.035	0.994	0.971

Hindi, Telugu, Tamil, Marathi, Malayalam, Kannada, Urdu, Gujrati and Odia. The dataset had 10-s long clips, and our system produced an accuracy of 95.52% whose confusion matrix is presented in Table 27. We had computed different performance metrics for this dataset using our proposed technique which are presented in Table 28.

It is observed from Table 27 that 78 clips from English were classified as Urdu. The misclassified instances were

Table 29 Obtained accuracies using our proposed technique for the top-10 world languages

	Chinese	Spanish	English	Arabic	Hindi	Bangla	Portuguese	Japanese	Punjabi	Russian
Chinese	2255	0	33	34	0	12	30	34	5	48
Spanish	1	2326	17	10	4	2	0	1	8	0
English	4	24	2181	53	33	4	5	26	30	24
Arabic	10	30	24	2212	25	11	13	31	19	27
Hindi	2	10	16	3	2312	23	13	19	7	1
Bangla	3	14	6	4	35	2349	14	18	9	6
Portuguese	16	2	6	3	11	11	2313	23	10	22
Japanese	7	15	20	15	11	9	17	2152	6	17
Punjabi	1	17	11	4	7	4	5	5	2339	1
Russian	3	0	2	11	2	2	5	13	1	2438

Table 30 Class-wise performance metrics for the top-10 world languages

	Sensitivity	Specificity	Precision	Negative predictive value	False-positive rate	False discovery rate	False-negative rate	Accuracy	*F*1 score
Chinese	0.98	0.991	0.92	0.998	0.009	0.08	0.02	0.989	0.949
Spanish	0.954	0.998	0.982	0.995	0.002	0.018	0.046	0.993	0.968
English	0.942	0.99	0.915	0.994	0.01	0.085	0.058	0.985	0.928
Arabic	0.941	0.991	0.921	0.993	0.009	0.079	0.059	0.986	0.931
Hindi	0.948	0.995	0.961	0.994	0.005	0.039	0.052	0.99	0.954
Bangla	0.968	0.995	0.956	0.996	0.005	0.044	0.032	0.992	0.962
Portuguese	0.958	0.995	0.957	0.995	0.005	0.043	0.042	0.991	0.957
Japanese	0.927	0.994	0.948	0.992	0.006	0.052	0.073	0.988	0.937
Punjabi	0.961	0.997	0.977	0.995	0.003	0.023	0.039	0.993	0.969
Russian	0.943	0.998	0.984	0.993	0.002	0.016	0.057	0.992	0.963

analyzed, and it was found that most of the clips had background music which was similar to those of the Urdu clips. It was also observed that in disparate instances both these languages had utterance of names of places and people which further added to this confusion.

The second dataset comprised of over 70 h of data from the top-10 spoken languages of the world [46]. The languages embodied were Chinese, Spanish, English, Arabic, Hindi, Bangla, Portuguese, Japanese, Punjabi and Russian. We obtained an accuracy of 95.21% with the proposed CNN-based technique. The confusion matrix for the same is presented in Table 29. It is observed that English and Arabic clips were confused with each other in some instances. The clips leading to such confusions were analyzed, and it was found that most of them were from news correspondents within open air condition. The quantity of noise was much higher as compared to the speech signals, and in many instances the speech was barely audible.

It is also observed that 58 clips were confused among the Bangla–Hindi pair. On analysis, it was found that in several interviews of Bangla, the speakers used Hindi words every now and then. Moreover, they also referred to names of places and famous people which led to such confusion. In the case of Hindi interviews, speakers often used phrases in English in addition to keywords which led to some of the Hindi clips being misclassified as English and vice versa. Different performance metrics were also calculated for this dataset using our proposed method which are presented in Table 30.

Both the datasets were subjected to the MFCC [21] and LSP-based [13] features along with different classifiers whose results are presented in Table 31.

6 Conclusion

Language identification is an important task for speech recognition in multilingual scenario. In this paper, a deep learning-based language identification system is presented which works by visualizing sound textures of each language with the aid of spectrograms. Experiments were performed for seven Indic languages from the standard

Table 31 Accuracies (%) of different classifiers on the two datasets with MFCC [21] and LSP-based [13] features

	Indian Top-11		World Top-10	
	MFCC [21]	LSP [13]	MFCC [21]	LSP [13]
BayesNet	67.34	78.49	64.17	72.51
Naïve Bayes	58.32	50.03	50.76	47.53
RBF Net	56.29	67.58	52.14	63.66
SVM	85.79	53.88	82.1	58.63
MLP	80.78	82.44	60.25	76.64
Lib	66.65	37.62	70.6	42.03
RF	89.91	87.44	87.41	88.85
CNN	**95.52**		**95.21**	

dataset (The IIIT-H Indic Speech Databases). The system is capable of detecting five out of the seven languages with 100% accuracy and the rest with over 99% accuracy. The system's performance was compared with reported works in the literature, popular audio and textural features, and it produced results with minimal error. Different performance metrics were computed for our proposed system as well as other systems which point to support our better results. The system's tolerance to noise was tested, and for a relative decrease of 4018.60% in the SNR, a decrease of only 0.50% in the system accuracy was observed which points toward its robustness.

As our system works on a GPU-based architecture, our immediate plan is to develop an architecture that does not require GPU for execution. Besides, the system will need to be equipped with real-time audio processing capability, where active learning [47] will be taken into account. We will also use voice activity detection [39] to further improve the performance of our system. Experimentation with clustering [48]-based techniques for unsupervised learning is our another plan. We also encompass the use of other standard image classification techniques [49, 50] and feature selection techniques [51] that provide a new scope in the domain.

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest.

References

1. Pan S-T, Lan M-L (2014) An efficient hybrid learning algorithm for neural network-based speech recognition systems on FPGA chip. Neural Comput Appl 24(7–8):1879–1885
2. Mustafa MK, Allen T, Appiah K (2019) A comparative review of dynamic neural networks and hidden Markov model methods for mobile on-device speech recognition. Neural Comput Appl 31(2):891–899
3. Jun S, Kim M, Oh M, Park H-M (2013) Robust speech recognition based on independent vector analysis using harmonic frequency dependency. Neural Comput Appl 22(7–8):1321–1327
4. Dua M, Aggarwal R, Biswas M (2018) Discriminatively trained continuous Hindi speech recognition system using interpolated recurrent neural network language modeling. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3499-9
5. Dudley WH (1939) The vocoder. Bell Labs Rec 18:122
6. Mukherjee H, Halder C, Phadikar S, Roy K (2017) Read—a Bangla phoneme recognition system. In: Proceedings of the 5th international conference on frontiers in intelligent computing: theory and applications. Springer, pp 599–607
7. Tang Z, Wang D, Chen Y, Shi Y, Li L (2017) Phone-aware neural language identification. In: 2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA). IEEE, pp 1–6
8. Giwa O, Davel MH (2017) The effect of language identification accuracy on speech recognition accuracy of proper names. In: 2017 Pattern recognition association of South Africa and robotics and mechatronics (PRASA-RobMech). IEEE, pp 187–192
9. Gunawan TS, Husain R, Kartiwi M (2017) Development of language identification system using MFCC and vector quantization. In: 2017 IEEE 4th international conference on smart instrumentation, measurement and application (ICSIMA). IEEE, pp 1–4
10. Masumura R, Asami T, Masataki H, Aono Y (2017) Parallel phonetically aware DNNS and LSTM-RNNS for frame-by-frame discriminative modeling of spoken language identification. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5260–5264
11. He J, Zhang Z, Zhao X, Li P, Yan Y (2016) Similar language identification for Uyghur and Kazakh on short spoken texts. In: 2016 8th international conference on intelligent human–machine systems and cybernetics (IHMSC), vol 2. IEEE, pp 496–499
12. Jin M, Song Y, McLoughlin I, Dai L-R (2018) LID-senones and their statistics for language identification. IEEE/ACM Trans Audio Speech Lang Process 26(1):171–183
13. Mukherjee H, Obaidullah SM, Phadikar S, Roy K (2018) A Dravidian language identification system. In: 2018 24th international conference on pattern recognition (ICPR). IEEE, pp 2654–2657
14. Gupta M, Bharti SS, Agarwal S (2017) Implicit language identification system based on random forest and support vector machine for speech. In: 2017 4th international conference on power, control & embedded systems (ICPCES).IEEE, pp 1–6
15. Madhu C, George A, Mary L (2017) Automatic language identification for seven Indian languages using higher level features. In: 2017 IEEE international conference on signal processing, informatics, communication and energy systems (SPICES). IEEE, pp 1–6
16. Nercessian S, Torres-Carrasquillo P, Martinez-Montes G (2016) Approaches for language identification in mismatched environments. In: 2016 IEEE spoken language technology workshop (SLT). IEEE, pp 335–340
17. Rebai I, BenAyed Y, Mahdi W (2017) Improving of open-set language identification by using deep SVM and thresholding functions. In: 2017 IEEE/ACS 14th international conference on computer systems and applications (AICCSA). IEEE, pp 796–802
18. Berkling KM, Arai T, Barnard E (1994) Analysis of phoneme-based features for language identification. In: Proceedings of ICASSP'94. IEEE international conference on acoustics, speech and signal processing, vol 1. IEEE, pp I–289

19. Srivastava BML, Vydana H, Vuppala AK, Shrivastava M (2017) Significance of neural phonotactic models for large-scale spoken language identification. In: 2017 international joint conference on neural networks (IJCNN). IEEE, pp 2144–2151
20. Tang Z, Wang D, Chen Y, Li L, Abel A (2018) Phonetic temporal neural model for language identification. IEEE/ACM Trans Audio Speech Lang Process 26(1):134–144
21. Mukherjee H, Obaidullah SM, Santosh K, Phadikar S, Roy K (2019) A lazy learning-based language identification from speech using MFCC-2 features. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-019-00928-3
22. Mukherjee H, Dhar A, Phadikar S, Roy K (2017) RECAL—a language identification system. In: 2017 international conference on signal processing and communication (ICSPC). IEEE, pp 300–304
23. Watanabe S, Hori T, Hershey JR (2017) Language independent end-to-end architecture for joint language identification and speech recognition. In: 2017 IEEE automatic speech recognition and understanding workshop (ASRU). IEEE, pp 265–271
24. Revathi A, Jeyalakshmi C, Muruganantham T (2018) Perceptual features based rapid and robust language identification system for various Indian classical languages. In: Computational vision and bio inspired computing. Springer, pp 291–305
25. Zissman MA, Singer E (1994) Automatic language identification of telephone speech messages using phoneme recognition and n-gram modeling. In: Proceedings of ICASSP'94. IEEE international conference on acoustics, speech and signal processing, vol 1. IEEE, pp I–305
26. Zissman MA (1995) Language identification using phoneme recognition and phonotactic language modeling. In: 1995 international conference on acoustics, speech, and signal processing, vol 5. IEEE, pp 3503–3506
27. Saikia R, Singh SR, Sarmah P (2017) Effect of language independent transcribers on spoken language identification for different Indian languages. In: 2017 international conference on Asian language processing (IALP). IEEE, pp 214–217
28. Lamel LF, Gauvain J-L (1993) Cross-lingual experiments with phone recognition. In: 1993 IEEE international conference on acoustics, speech, and signal processing, vol 2. IEEE, pp 507–510
29. Ghozi R, Fraj O, Jaïdane M (2007) Visually-based audio texture segmentation for audio scene analysis. In: 2007 15th European signal processing conference. IEEE, pp 1531–1535
30. Dennis JW. Sound event recognition in unstructured environments using spectrogram image processing. Nanyang Technological University, Singapore
31. Montalvo A, Costa YM, Calvo JR (2015) Language identification using spectrogram texture. In: Iberoamerican congress on pattern recognition. Springer, pp 543–550
32. Prahallad K, Kumar EN, Keri V, Rajendran S, Black AW (2012) The IIIT-H Indic speech databases. In: Thirteenth annual conference of the international speech communication association
33. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436
34. Zhang D, Han X, Deng C (2018) Review on the research and practice of deep learning and reinforcement learning in smart grids. CSEE J Power Energy Syst 4(3):362–370
35. Sang J, Yu J, Jain R, Lienhart R, Cui P, Feng J (2018) Deep learning for multimedia: science or technology? In: Proceedings of the 2018 ACM multimedia conference on multimedia conference, ACM, pp 1354–1355
36. Olivas-Padilla BE, Chacon-Murguia MI (2019) Classification of multiple motor imagery using deep convolutional neural networks and spatial filters. Appl Soft Comput 75:461–472
37. Chevtchenko SF, Vale RF, Macario V, Cordeiro FR (2018) A convolutional neural network with feature fusion for real-time hand posture recognition. Appl Soft Comput 73:748–766
38. Wang Y, Chen Y, Yang N, Zheng L, Dey N, Ashour AS, Rajinikanth V, Tavares JMR, Shi F (2019) Classification of mice hepatic granuloma microscopic images based on a deep convolutional neural network. Appl Soft Comput 74:40–50
39. Mukherjee H, Obaidullah SM, Santosh K, Phadikar S, Roy K (2018) Line spectral frequency-based features and extreme learning machine for voice activity detection from audio signal. Int J Speech Technol 21(4):753–760
40. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11(1):10–18
41. Mohanaiah P, Sathyanarayana P, GuruKumar L (2013) Image texture feature extraction using GLCM approach. Int J Sci Res Publ 3(5):1
42. Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
43. Chen J, Shan S, He C, Zhao G, Pietikainen M, Chen X, Gao W (2009) WLD: a robust local image descriptor. IEEE Trans Pattern Anal Mach Intell 32(9):1705–1720
44. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
45. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(Jan):1–30
46. Simons GF, Fennig CD (2017) Ethnologue: languages of Asia. SIL International, Dallas
47. Bouguelia M-R, Nowaczyk S, Santosh K, Verikas A (2018) Agreeing to disagree: active learning with noisy labels without crowdsourcing. Int J Mach Learn Cybern 9(8):1307–1319
48. Bhattacharyya S, Snasel V, Dey A, Dey S, Konar D (2018) Quantum spider monkey optimization (QSMO) algorithm for automatic gray-scale image clustering. In: 2018 international conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 1869–1874
49. Nath SS, Mishra G, Kar J, Chakraborty S, Dey N (2014) A survey of image classification methods and techniques. In: 2014 international conference on control, instrumentation, communication and computational technologies (ICCICCT). IEEE, pp 554–557
50. Duda RO, Hart PE, Stork DG (2012) Pattern classification. Wiley, Hoboken
51. Das AK, Sengupta S, Bhattacharyya S (2018) A group incremental feature selection for classification using rough set theory based genetic algorithm. Appl Soft Comput 65:400–411